# A Design of Web Log Integration Framework Using NoSQL

Huijin Jeong[1], Junho Choi[1], Chang Choi[1], Ilsun You[2], and Pankoo Kim[1,*]

[1] Department of Computer Engineering Chosun University,
375 Seoseok-dong, Dong-gu, Gwangju, Republic of Korea
{Jeonghuijin,enduranceaura}@gmail.com,
xdman@paran.com, pkkim@chosun.ac.kr
[2] School of Information Science Korean Bible University,
16 Danghyun 2-gil, Nowon-gu, Seoul, Republic of Korea
isyou@bible.ac.kr

**Abstract.** Webservice is a software technology as the representative method of information communication currently used to create a dynamic system environment that is configured to fulfill its users' needs. Therefore, analyzing log data that occurred at provision is being used as the significant basic data in webservice research. Thanks to development of Cloud computing technology, it has resulted in centralized points from which data is generated and data enlargement. A research is now implemented to create information from collecting, processing and converting flood of data and to obtain the new various items of information. Against this backdrop, it is justified that collection, storage and analysis of web log data in the existing conventional RDBMS system may be inadequate to process the enlarged log data. This research propose a framework which to integrate web log for storage using HBase, a repository of the Cloud computing- based NoSQL. In addition, data validation must be completed in the pre-process when collecting web log. The validated log is stored in the modeling structure in which takes features of web log into account. According to the results, introduction of NoSQL system is found to integrate the enlargement of log data in more efficient manner. By comparisons with the existing RDBMS in terms of data processing performance, it was proved that the NoSQL- based database had a superior performance.

**Keywords:** Big Data, Security Log aggregation, Cloud computing, NoSQL, HBase.

## 1 Introduction

Webservice can be the most representative service in which a business and customers communicate each other. Analysis of web log data from its operation allows for re-processing various information that include status of a system, popular service pages after being extracted into any contributing information to the operation [15].

* Corresponding author.

Linawati et al. (Eds.): ICT-EurAsia 2014, LNCS 8407, pp. 438–445, 2014.

Nonetheless, as Smart Phone penetration increases in recent times, this has made it possible to access to the service regardless of time and place, constantly increasing the amount of log. Because of this, it brings about the newly-coined word of Big Data [3]. There have been vigorous research efforts in various fields to derive different type of information from Big Data by the process of understanding [1, 16, 4, 18, 19], analyzing and gathering. In general, the existing web log repository consists of RDBMS. However, due to data enlargement, there are many difficulties to manually analyze web log and efficiently manage the service. RDBMS is not enough to process such gigantic data completely. The most representative technology that overcomes the incompetence is referred to as a distributed database or NoSQL. The most comparative advantages with NoSQL include that it supports for the horizontal expansions which enable massive data to be processed to overcome the limitations. In addition, it has strength in not using schema that presents a relationship data hold. As a result, a web log with different structures can smoothly be replaced with a log of the new formation [2]. The propose framework is a formalized one that fulfills the objective for collecting, storing and analyzing the massive web log data using HBase, a repository of NoSQL data in effective manner. Under the assumption that different web logs are integrated, a Cloud computing environment was considered, and a web log which had little effect on the existing system at storage and at the same time was configured to function for collection in real time was through preprocessing for eliminating unnecessary data after taking field structures into account. Later, data modeling structure for NoSQL environment was designed in order to store data. To verify the framework that is being proposed, it checked whether the massive web log data was more speedily processed using Wikipedia Accesslog Dataset [8] than in the previously used RDBMS. the reminder of the research is organized as follows: Chapter 2 it describes NoSQL-based log integration with comparisons between its features and log repository. Chapter 3 describes a preprocessing in a way that efficiently integrate a web log with the propose web log integration framework as well as a HBase-based modeling structure for integration. In chapter 4 an experiment of data processing speed and its results, as a solution that is proposed in chapter 3 are described. In the final chapter 5 it ends with conclusions and future research directions.

## 2 Related Works

Log data refers to recording information that includes sequential events occurred in system operation as well as details of operation in system or on network [17]. Logs can be classified into operating system, network packet logs, and internet access record log. The existing RDBMS is designed with the standards of integrity and consistency of data, and it stores log data. RDBMS has an advantage in that it designs a table with normalized schema, proving join functions and with data expression opened. In contrast, the formalized schema structure has a significant difficulty in database expansions as it is not proper for the distributed processing environment. The supplementary database for expansions is developed and called as NoSQL(Not Only SQL) [5, 6]. Right off the bat, NoSQL databases are unique because they are usually independent from Structured Query Language (SQL) found in relational databases. Relational databases all use SQL as the domain-specific language for ad hoc

queries, while non-relational databases have no such standard query language, so they can use whatever they want. That can, if need be, include SQL [14]. In addition, it promises no perfect data consistency as the distributed storage database, by its characteristics, is focused on system provision even though a few database with data storage fail to respond [7, 12]. There has been an ongoing research in which NoSQL is used as repository to effectively process the ever-growing data [13]. NoSQL repository is divided into 4 types of data storage structures. Out of the four structures, HBase which opts for Column Model is usefully employed as the non-formalized data repository since it is capable of configuring multi-dimensional data based on columns [9]. Additionally, with employment of multi-slave method when configuring data nodes, it can flexibly be used in log integration.

Choi et al. [10] had proposed security log analysis system with a NoSQL-based Mapreduce design that allowed firewall log data of this type with higher capacity to be collected and analyzed for integration, compared to RDBMS in terms of data processing performance and performed an analysis of the three attack patterns selected for evaluation.

WEI et al. [11] had proposed a system using MongoDB, a repository of NoSQL in order to integrate a large number of networks monitoring log data. The research included a system design that took respective features of hardware layers and application layers in Cloud environment into consideration, Mapreduce programming that worked for log integration processing of massive log data in effective manner, and the improvement of log integrated architecture compared to RDBMS.

## 3 Hbase-BASED Log Integration

### 3.1 Proposed Framework

The integration of security logs must not create any problems with the service, even if the security logs often occur such as, added, modified and deleted thing. Also, these security logs shall be no degradation in system performance through big security logs for detecting security incident. Therefore, this paper proposes an integrated system for good storage efficiency based on big log data. The Figure 1 is a framework of the proposed log integration system.

There are some preconditions for the configuration of log analysis system as follows:

(1) System operation and assuming that the attack happens.
(2) The web logs are created in the working service.
(3) Confirm to create the SSH Tunnel for transmission of web logs.
(4) Transfer the variety web log data through web log collector.
(5) Perform the preprocessing of transferred web logs.
(6) The web log data are stored in HBase after preprocessing.
(7) The Extraction of pattern and information using stored web log data.
(8) The extracted information is forwarded to the administrator.

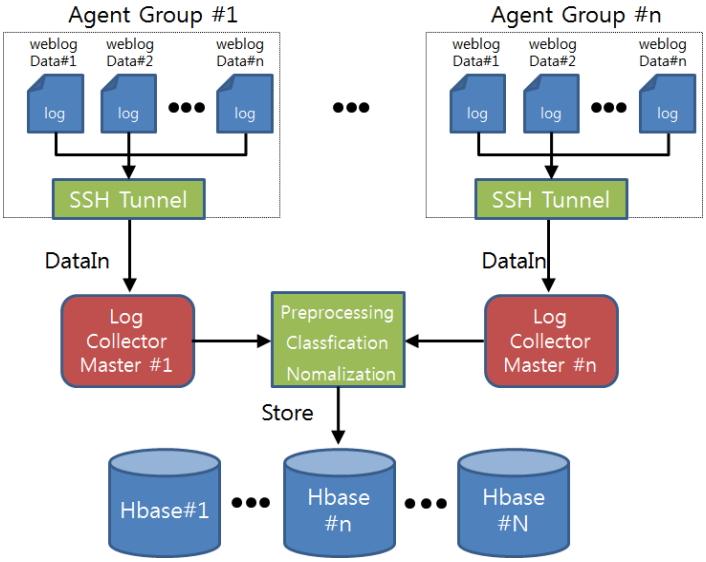


**Fig. 1.** Framework of the proposed log integration system

The first step is the collection of big logs for security logs integration. The security logs are transferred through SSH Tunnel and these are performed the preprocessing for removing of security threats. The reason of performing preprocessing are a step for removing unnecessary information after checking the data type of a heterogeneous logs. Also, there is needed for the model structure according to the type HBase after extracting necessary data. Finally, the integrated log storage when storing the common purpose is to integrate information.

### 3.2 Web Log Preprocessing

The second step is the preprocessing of big log data. The preprocessing is classified the format type after checking the contents of log data. Also, this step is reduced the waste of storage space by removing unnecessary data. The type classification is very important because the structure of log data is different in occurred log data of heterogeneous services. The figure 2 is a framework of the preprocessing.

There are some steps of preprocessing as follows:

(1) The variety web log data is occurred in service.
(2) To validate the input data as a web log.
(3) To classify through analysis of web log data.
(4) To format by administrator using the characteristics of classified web log.
(5) The purified data is stored.
(6) Repeat steps 1 to 6 times.

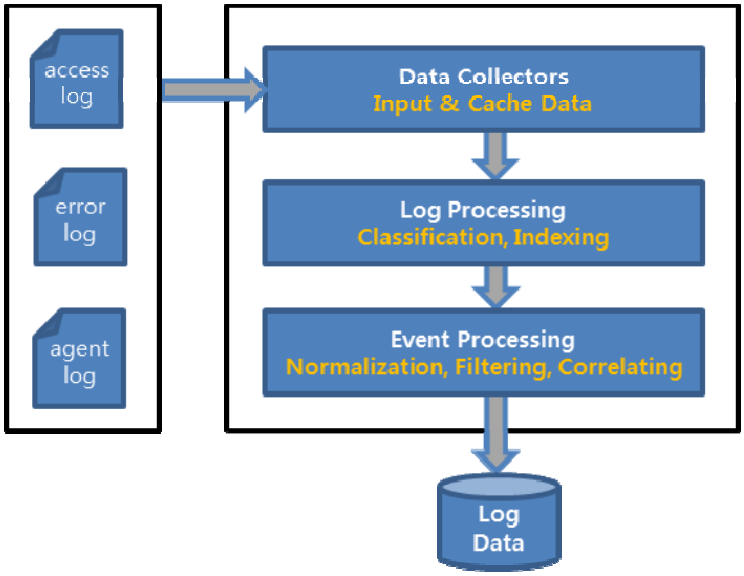


**Fig. 2.** Framework of the Preprocessing

### 3.3 Data Modeling for Log Integration

Amount of large scaled security logs which should be collected and stored under cloud computing environment is rapidly increasing. However, conventional collection method based on RDBMS storage can't afford the amount of security logs. In order to solve this problem, this paper proposes NoSQL-based method to collect and integrate security logs. NoSQL is more effective and rapid data storage than conventional RDBMS. Actually, data modeling based on relational database can performs freely queries through setting relation between tables. However, the design of data store model is needed because NoSQL does not support complex queries.

The Figure 3 is an example of the log integrated modeling framework. In figure 3, the Host table is entered 'Host' information using Row Key for data identification from LogData table in structure of the framework. Also, data is generated by the column such as, Log the type, log generation time, count and so on. The type of web log is classified by time in service. The time information is the most basic and important information for security incident or analysis of operational status because the data is stored by time sequentially. The Row key is defined 'DateTime' + 'Host' + 'Count' information in Log-Data table for log storage because of data query information such as, 'who', 'when', 'occurrence' and so on. DataLogInfo Column is classified Accesslog, Errorlog and Agentlog by log characteristics.

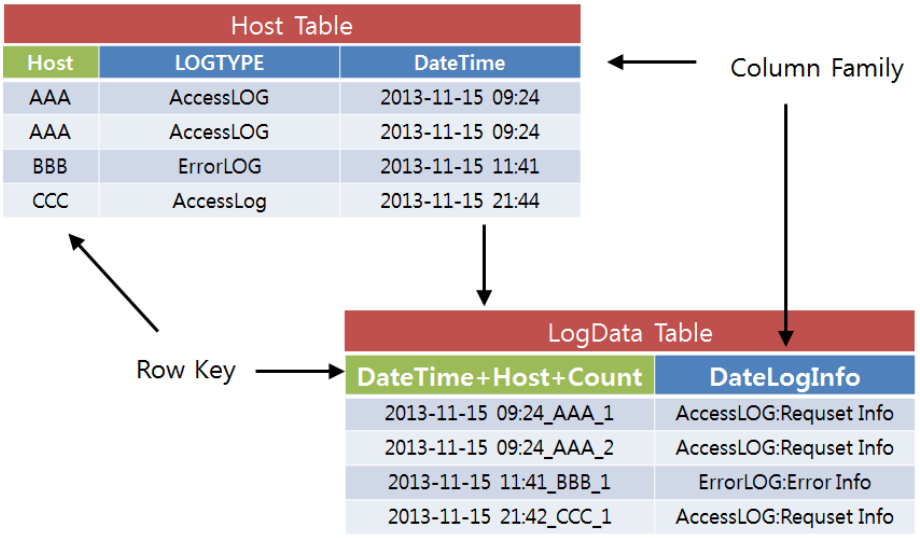


**Fig. 3.** Framework of log integration modeling

# 4 Performance Evaluation

## 4.1 Experimental Environment

The performance evaluation of data input is performed between NoSQL and RDMS. RDMS is MySQL-5.1.69 because the most widely used. NosQL is used HBase-0.92.12 and it consists of master node, 4 data nodes, 50 Thread. The experiment is performed the data input performance using Wikipedia AccessLog data set [8]. The AccessLog data set based on Wikipedia are consists of 2 million, 5 million and 10 million elements.

The Figure 4 is the result of performance test. This test is processed number per hour and integration available amount based on web log.

## 4.2 Evaluation

In the results, 2mil is not large difference count but 5mil and 10mil can be clearly confirmed large difference count between HBase and MySQL. Also, the performance can be improved by increasing Data Node. If number of DataNodes is ensured sufficient, it seems possible additional performance boost.

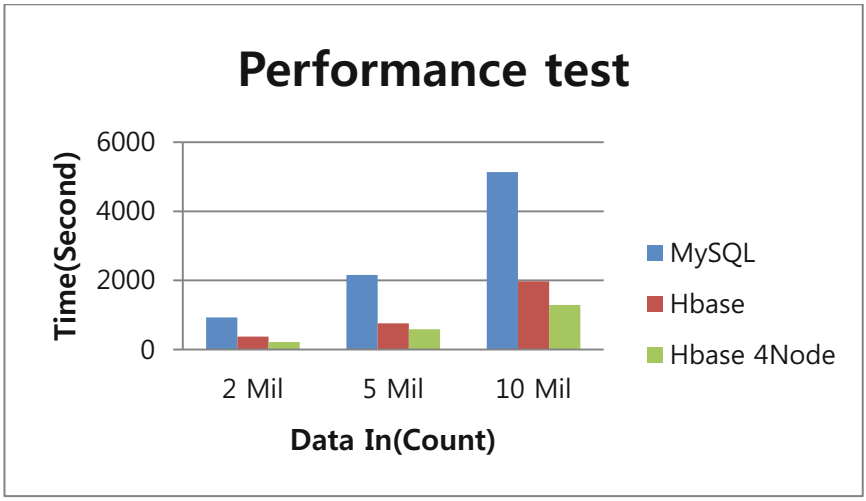


**Fig. 4.** RDBMS VS Hbase

## 5 Conclusion and Future Works

As cloud computing technologies are rapidly advancing, cloud environment is significantly expanding. Although cloud environment provides users with convenience, prevention and detection of possible security invasion accidents is still unsolved problems. The most intrinsic method to prevent security invasion accident is to collect security logs for each system and then analyze them.

This paper proposes NoSQL-based large capacity security log integration method for cloud security platform. Since cloud computing provides various services to users, a new web log that is different from existing one is likely to occur. Therefore, this paper proposes large scaled web log management to collect, store and integrate logs considering characteristics between heterogeneous machines. In the future, it needs the effective analysis methods based on heterogeneous web logs.

**Acknowledgments.** This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2013R1A1A2A10011667).

## References

1. Choi, J., Choi, C., Ko, B., Choi, D., Kim, P.: Detecting Web based DDoS Attack using MapReduce operations in Cloud Computing Environment. Journal of Internet Services and Information Security 3(3/4), 28–37 (2013)
2. Oliner, A., Ganapathi, A., Xu, W.: Advances and challenges in log analysis. Communications of the ACM 55(2), 55–61 (2012)

3. Yunhua, G.U., Shu, S., Guansheng, Z.: Application of NoSQL Database in Web Crawling. International Journal of Digital Content Technology and its Applications 5(6) (2011)
4. Elkotob, M., Andersson, K.: Cross-Layer Design for Improved QoE in Content Distribution Networks. IT CoNvergence PRActice (INPRA) 1(1), 37–52 (2013)
5. Srinivasan, V., Bulkowski, B.: Citrusleaf: A Real-Time NoSQL DB which Preserves ACID. In: The 37th International Conference on Very Large Data Bases, Proceedings of the VLDB Endowment, vol. 4(12), pp. 1340–1350 (2011)
6. Yi, X., Wei, G., Dong, F.: A Survey on NoSQL Database. Communication of Modern Technology, 46–50 (2010)
7. Zhou, W., Han, J., Zhang, Z., Dai, J.: Dynamic Random Access for Hadoop Distributed File System. In: 32nd International Conference on Distributed Computing Systems Workshops, pp. 17–22. IEEE (2012)
8. http://www.wikibench.eu
9. George, L.: HBase The Definitive Guide. O'ReillyMedia (2011)
10. Bomin, C., Jong-Hwan, K., Sung-Sam, H., Myung-Mook, H.: The Method of Analyzing Firewall Log Data using Map Reduce based on NoSQL. Journal of The Korea Institute of Information Security & Cryptology (JKIISC) 23(4), 667–677 (2013)
11. Yang, J., Leskovec, J.: Patterns of Temporal Variation in Online Media. In: ACM International Conference on Web Search and Data Mining, pp. 177–186 (2011)
12. Shvachko, K.V.: HDFS Scalability: The limits to growth. Login 35(2), 6–16 (2010)
13. Borthakur, D., Sarma, J.S., Gray, J.: Apache Hadoop goes realtime at Facebook. In: SIGMOD, pp. 1071–1080 (2011)
14. Proffitt, B.: When NoSQL Databases Are Yes Good For You and Your Company (2013)
15. Agosti, M., Crivellari, F., Nunzio, G.M.: Web log analysis: a review of a decade of studies about information acquisition, inspection and interpretation of user interaction. Data Mining and Knowledge Discovery 24(3), 663–696 (2012)
16. Choi, C., Choi, J., Ko, B., Oh, K., Kim, P.: A Design of Onto-ACM (Ontology based Access Control Model) in Cloud Computing Environments. Journal of Internet Services and Information Security 2(3/4), 54–64 (2012)
17. Herrerias, J., Gomez: Log Analysis Towards an Automated Forensic Diagnosis System. IEEE ARES, 15–18 (2010)
18. Han, S., Han, Y.: Meaning and Prospects of IT Convergence Technology in Korea. IT CoNvergence PRActice (INPRA) 1(1), 2–12 (2013)
19. Gonzalez-Miranda, S., Alcarria, R., Robles, T., Morales, A., Gonzalez, I., Montcada, E.: An IoT-leveraged information system for future shopping environments. IT CoNvergence PRActice (INPRA) 1(3), 49–65 (2013)